BT2 - Heuristics & A\* search

Chào cả lớp,

Chúng ta tiếp tục giải các game Sokoban như trong BT1. Trong bài tập này chúng ta cần cài đặt thuật toán A\* và so sánh hiệu suất của A\* với UCS.

**Một số lưu ý như sau:**

* Hiện tại trong code mẫu đính kèm đã cài sẵn một hàm heuristic. Các em có thể sử dụng hàm heuristic này hoặc tự thiết kế các heuristic khác tốt hơn.
* Hoàn thành cài đặt thuật toán A\* trong hàm **aStarSearch**trong file**solver.py**. (Lưu ý: Có thể tự code lại sử dụng cấu trúc dữ liệu tự định nghĩa).
* Để thay đổi thuật toán chạy thì cần comment/uncomment các dòng lệnh tương ứng trong hàm **auto\_move()** trong file **game.py.**
* Cài đặt đo thời gian chạy của mỗi thuật toán (UCS và A\*). (Gợi ý: Sử dụng hàm time() ở đầu và cuối mỗi thuật toán để đo xem thuật toán cần bao nhiêu thời gian (tính theo số giây) để giải mỗi màn chơi).

**Bài nộp gồm có 2 file:**

* 1 file **zip**chứa source code trong đó cài đặt UCS và A\* với comment chú thích đầy đủ cho các dòng lệnh của thuật toán. Đặt tên BT2\_MSSV.zip với MSSV là mã số sinh viên.
* 1 file **pdf**báo cáo trong đó cần trình bày lại heuristics đã sử dụng có ý tưởng là gì? Nếu bạn nào thiết kế và thực nghiệm so sánh thêm với **heuristics mới** thì sẽ được cộng điểm *(Cần phải so sánh với heuristics đang có sẵn trong code mẫu)*.
* Trong file báo cáo cần có **Bảng thống kê**so sánh **thời gian chạy** (số giây) và **số nút** đã được mở ra của hai thuật toán UCS và A\* cho mỗi level. Cần trả lời câu hỏi: Lời giải trả về của A\* có phải lời giải tối ưu hay không cho level tương ứng.

**Lưu ý:**

* File báo cáo không phải là pdf sẽ trừ 10/100 điểm.
* Báo cáo phải viết cẩn thận, rõ ràng, mạch lạc. Cần tự thiết kế cách trình bày sao cho gọn gàng, sạch sẽ. Không nên dài quá 10 trang.
* Tất cả bài làm có dấu hiệu sao chép sẽ nhận 0 điểm.

**Deadline: 31/3/2024.** *Sau ngày 3/4/2024 nộp bài sẽ bị trừ điểm dần dần. Sau ngày 14/4/2024 sẽ không nhận thêm bài nộp mới.*